

# A Recursive Exponential Filter For Time-Sensitive Data

LAUR-99-5573

Phillip D. Stroud

© October 8, 1999

Los Alamos National Laboratory  
Mail Stop F607, Los Alamos, NM 87544  
stroud@lanl.gov

**Abstract** – A recursive formulation of an exponential smoothing filter is developed, within the framework of a least square error approach with data uncertainties that increase exponentially with time. An efficient implementation into Java is presented. By analogy to the Kalman filter, an interpretation of the gain as a ratio of uncertainties leads to a measure of validity for the recursive exponential filter. The time sensitive recursive exponential filter is then used in a detection/classification application in a natural environment with non-stationary process statistics (the concentration and size distribution of atmospheric aerosols). The performance is found to be superior to a non-adaptive Kalman filter and to a moving average filter.

## ***1. Introduction***

In decision-making or control systems, it is a common need to know when a value underlying a stream of noisy data has undergone a significant change. Of particular interest is determining whether an event or transition with a known characteristic time scale has occurred. In this case, longer time scale changes must be ignored because they would not be attributable to the phenomena of interest, while shorter time scale changes are attributed to high-frequency process noise, and should be ignored. In addition, observation noise must be filtered out. The motivating application is to recognize when an artificial release of aerosol has been introduced into the ambient atmosphere, based on a stream of noisy concentration observations.

The effects of short term noise and observation noise (shot noise) can be reduced by applying smoothing filters to the data stream. Such a filter continually calculates a smoothed value given by a weighted average of recent observations. Older observations receive less weight. Such smoothing filters can come in many possible “shapes”, including rectangular (which generates moving averages), exponential, Gaussian, linear ramp, power law, etc. These filters have an associated time scale, characterized by how old an observation would have to be for it to receive one half or  $1/e$  of the weight attributed to a brand new observation. If there are  $N$  observations during this time period, the filter will

reduce by a factor of approximately  $N$  the variance of fluctuations on the signal of interest due to observation noise and high-frequency process noise.

The effects of long term process noise, or non-stationarity of the process, also need to be removed from the signal if the phenomena of interest is known to have an associated time scale. When looking for artificial aerosol releases, signal changes in a few minute time scale are of interest, while concentration rises that occur over a half hour are probably due to atmospheric phenomena (long term process noise). The effect of these long term process noises can be removed from the signal by subtracting a long-term filter from a shorter term filter.

A recursive implementation of a filter eliminates the need to store and reprocess many old data records each time the filter is updated. In a recursive filter, the best-estimate of the underlying value is kept. When a new datum arrives, the new best estimate is obtained by a linear combination of the previous value with the new observation. A recursive filter is less degraded by data drop outs than other more rigid filter implementations. An exponential filter can be implemented in a recursive form.

Because of obvious structural similarities, the recursive exponential filter can be interpreted within the Kalman formulation. This analogy leads to a measure of validity for the values produced by the recursive exponential filter. This knowledge of validity can support decisions made with the filtered values.

## 2. General Best Estimate Formulation and the Exponential Filter

Suppose there is a set of  $n$  values  $\{z_i\}$  each representing an underlying value  $u$  with uncertainty  $\sigma_i^2$ . Note that  $z$  can be the raw data values or the result of some operation on the data, such as a linear combination of data elements. It can be easily demonstrated [Press] that the value  $X$  which minimizes the certainty-weighted mean square fitting error  $\chi^2 = \frac{1}{N} \sum_i (X - z_i)^2 / \sigma_i^2$  is given by

$$X = \frac{\sum_i (z_i / \sigma_i^2)}{\sum_i (1 / \sigma_i^2)}. \quad (1)$$

The “best estimate”  $X$  is thus a weighted average of each representation of  $u$ , where each value is weighted by its certainty. The uncertainty with which the best estimate  $X$  represents the true value  $u$  is then found to be given by

$$\sigma_X^2 = \frac{1}{\sum_i 1 / \sigma_i^2}. \quad (2)$$

For the degenerate case in which all values represent the underlying value with the same certainty, Eq(1) gives a best estimate equal to the mean of all values, while Eq(2) gives the “variance in the mean” as the single value variance divided by the number of observations.

This formulation will next be used to develop a view of a data stream where the uncertainty associated with each datum depends on its age. This will allow treatment of situations where the environment or observed process is changing, causing older data to represent the current process state with less certainty.

Represent a data series observed to date as a sequence of pairs  $(z_i, t_i)$ , where  $z_i$  is the value obtained in observation  $i$ , which occurs at time  $t_i$ . The index  $i$

indicates sequential observations, with  $i+1$  following next after observation  $i$ . Up until time  $t_n$ , there are  $n$  observations. After observation  $n$ , the smoothed value produced by an exponential filter with time scale  $\tau$ , for times  $t$  after  $t_n$  is

$$x_n = \frac{\sum_{i=1}^n e^{-(t-t_i)/\tau} z_i}{\sum_{i=1}^n e^{-(t-t_i)/\tau}} \quad (3)$$

This value does not change with time, until the next datum is observed, because of the way the current time  $t$  appears in both numerator and denominator.

Comparison of Eq(3) with Eq(1) allows the interpretation that an exponential filter generates the best estimate of an underlying value under the following two conditions: 1) the uncertainty of all observations take the same value  $\sigma_0^2$  immediately following the measurement, and 2) the uncertainty of an observation increases exponentially with time, with a time scale of  $\tau$ . At time  $t$ , the uncertainty associated with observation  $i$ , which was taken at time  $t_i$ , is

$$\sigma_i^2(t) = \sigma_0^2 e^{(t-t_i)/\tau} \quad (4)$$

The uncertainty with which the exponential filter result represents the underlying value at time  $t$  is then, by substituting Eq(4) into Eq(2):

$$\sigma_x^2(t) = \sigma_0^2 / \sum_{i=1}^n e^{-(t-t_i)/\tau} \quad (5)$$

### 3. Recursive Formulation of the Exponential Filter

Eq(3) gives the result of the exponential filter after observation  $n$ . After the next observation,  $n+1$ , the value produced by the exponential filter is

$$x_{n+1} = \frac{\sum_{i=1}^n e^{-(t-t_i)/\tau} z_i + e^{-(t-t_{n+1})/\tau} z_{n+1}}{\sum_{i=1}^n e^{-(t-t_i)/\tau} + e^{-(t-t_{n+1})/\tau}} \quad (6)$$

With a little algebraic manipulation, the smoothed value after observation  $n+1$  can be cast as a linear combination of the smoothed value after observation  $n$  and the value obtained in observation  $n+1$ :

$$x_{n+1} = x_n + K_{n+1}(z_{n+1} - x_n), \quad (7)$$

as long as the gain,  $K$ , is given by

$$K_{n+1} = \frac{1}{1 + \sum_{i=1}^n e^{-(t_{n+1}-t_i)/\tau}}. \quad (8)$$

This gain can also be cast in a recursive form as

$$K_{n+1} = K_n / (K_n + e^{-(t_{n+1}-t_n)/\tau}). \quad (9)$$

The gain take values in (0,1). When no data has been received in quite a while (relative to the filter time scale), Eq.(9) shows that the gain for the next observation goes towards unity. In that case, according to Eq.(7), the prior estimate is disregarded when a new datum arrives, and the new estimate takes the value of the new observation.

On the other hand, when there are many observations during the filter's time scale, the gain will be small. For example, for a filter time scale of 60 seconds, with data arriving every 1 second, the steady state gain attained by Eq.(9) after a few minutes of uninterrupted data is approximately  $1/60$ . When the gain is small, the prior value is moved only slightly towards the new observation.

Data drop outs are much easier to handle in this recursive formulation than in non-recursive filter implementations. All that is needed to calculate the gain is the previous gain and the time interval between the current observation and the previous observation.

To begin filtering a stream of data,  $t_0$  is set to the distant past, and  $K_0$  is set to an arbitrary finite value. The gain to use for the first datum is given by Eq.(9) to be  $K_1 = 1$ . Then Eq.(7) gives that  $x_1 = z_1$ .

The simplicity of the recursive exponential filter can be seen in a Java implementation of the recursive exponential filter:

```
public class RecursiveExponentialFilter {
    private double x; // best estimate of underlying value
    private double K=1.0; // gain
    private double tau; // filter time scale
    private double tLast = -100000.0; // time of previous observation

    public RecursiveExponentialFilter(double _tau) { tau = _tau; }

    public void newObservation (double z, double t) {
        K = K / (K + Math.exp(-(t - tLast )/tau));
        x = x + K * (z - x);
        tLast = t;
    }

    public double getX() {return x;}
}
```

#### ***4. Interpretation of Recursive Exponential Filter within Kalman Framework***

In the Kalman formulation [Kalman, Kalman & Bucy], there is presumed to be a true but unknowable, underlying, hidden value,  $u(t)$ , of a process of interest. This underlying value changes with time according to the process dynamics. An observation of the underlying value gives a measured value that is corrupted by measurement error. (The Kalman formulation allows for a vector of observation variables and a vector of underlying state variables, linked by an observation matrix. For example, the underlying state of a satellite is a vector of seven orbit parameters, while the vector of observations may include measured elevation, azimuth, and time from a ground station. For simplicity, this analysis treats the case where an underlying state variable is measured directly, giving a one-to-one association between state variable and observation.) In addition, there is a formulated degradation in certainty of the best estimate with time.

The Kalman measurement equation relates the observed value  $z$  to the underlying value  $u$  by  $z_k = u_k + v_k$ , where the measurement noise  $v_k$  is a random variable with a variance of  $R$ . The subscript  $k$  indicates the  $k$ th measurement, which occurs at time  $t_k$ . The measurement noise at any observation is independent of that at any other measurement.

For stochastic processes, the change in the underlying value between subsequent observations is given by the Kalman state update equation:  $u_{k+1} = u_k + w$ , where  $w$  is the process noise, which has a variance  $Q$ , which is the expected value of the difference between subsequent underlying values, squared. The process noise variance characterizes the change in the underlying value in the time lapsing between observations.

There are three variables used to operate the Kalman filter: the filtered value  $x$ , the expected variance of the error  $P$ , and the gain  $K$ . For each observation  $k$ , these three variables have prior and posterior values.  $\hat{x}_k^-$  denotes the prior best estimate or filtered value associated with the underlying value, just before measurement  $k$ , while  $\hat{x}_k$  denotes just after.  $P_k^-$  denotes the estimated square of the error between  $\hat{x}_k^-$  and  $u_k$  just after observation  $k$  ( $P_k^-$  denotes just before.) In the multidimensional Kalman formulation,  $P$  is a covariance matrix rather than a single variance.  $K_k$  designates the gain used to update  $x$  after the  $k^{\text{th}}$  observation. The Kalman formulation gives a filtered value  $x$  that minimizes the expected error. Under assumptions of normally distributed noise, the filtered value  $x$  also represents the most likely underlying value.

There are five steps in updating the Kalman filter for each observation. The first two steps form the time-update or prediction part of the algorithm. The values  $\hat{x}_k^-$  and  $P_k^-$  just prior to observation  $k$  are obtained by updating the posterior values from just after the previous measurement. For a stochastic process,  $\hat{x}_k^-$  is simply equal to  $\hat{x}_{k-1}$ . The uncertainty with which  $x$  represents  $u_k$  increases during the time between observations  $k-1$  and  $k$ . The expected error increases, by an amount  $Q$  equal to the variance of the process during consecutive measurements:  $P_k^- = P_{k-1} + Q$ . For the first observation, the prior values  $\hat{x}_k^-$  and  $P_k^-$  are initialized by guessing.

The remaining three steps form the measurement update or correction part of the algorithm, that is executed when a new observation is made. First, the gain is calculated, using  $K_k = P_k^- / (P_k^- + R)$ . Then the filtered value is updated, using  $\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-)$ . Finally the variance of error is reduced by the new data:  $P_k = (1 - K_k) P_k^-$ .

$P$ ,  $Q$ , and  $R$  can be interpreted as uncertainties.  $P_k^-$  is the uncertainty with which  $\hat{x}_k^-$  represents  $u_k$ .  $R$  is the uncertainty with which observation  $z$  represents the underlying value  $u$ .  $Q$  is the uncertainty with which  $u_{k-1}$  represents  $u_k$ . The Kalman gain,  $K$ , is then the ratio of two uncertainties. The denominator is the sum of the uncertainty with which the prior estimate represents the underlying value and the uncertainty with which the observation represents the underlying value. The numerator is the uncertainty with which the prior estimate represents the next underlying value.

The measurement update equation for the recursive exponential filter, Eq(7) is, of course, identical to that of the Kalman formulation. The uncertainty with which the exponential filter represents the underlying value, given by Eq(5), can be interpreted as the Kalman expected variance of the error,  $P$ . The difference is that the update equation for the recursive exponential filter would be  $P_k^- = P_{k-1} \exp(-d/\tau)$  instead of the Kalman formulation  $P_k^- = P_{k-1} + Q$ . Here,  $d$  is the time between observation  $k-1$  and  $k$ .

The Kalman observation noise variance,  $R$ , can be identified with  $\sigma_0^2$ , the uncertainty associated with an observation immediately after it is made. With

these associations, the recursive formulation of the gain of the exponential filter, given by Eq(9), is identical to that obtained by the Kalman gain equation.

### **5. Validity of the Exponential Filter**

There are two measures that can be applied to characterize the validity of the filtered value. The first measure ensures that the uncertainty with which the filtered value represents the underlying value is sufficiently low. There are several ways to characterize this uncertainty. The approach taken here is to use the filtered value itself to determine whether the data is statistically significant. If the filtered value is below a threshold, it can be considered to be dominated by shot noise, and will be treated as invalid.

A second measure of validity of the recursive filter can be constructed from the gain. The lower the value of the gain, the more certainly does the filter value represent the current underlying value relative to the certainty with which the next observation will represent it. As with the Kalman filter, a minimum gain criteria can be used to determine whether sufficient data has been collected sufficiently recent for the value produced by the filter to be valid. The minimum gain can be formulated in terms of the equivalent consecutive number of observations that must be successfully made before the filtered value will be considered valid for use in decisions. If some recent data is missing, the filter may still produce a valid value if there is more, somewhat older, data. The gain threshold for validity of the filter requiring the equivalent of  $N$  consecutive observations taken at intervals of  $\Delta t$  would then be

$$K_{\max} = \frac{1 - e^{-\delta t / \tau}}{1 - e^{-N \delta t / \tau}} \quad (10)$$

As long as the gain produced by Eq(9) is lower than this threshold, the filtered value can be considered valid.

### **7. Application**

There are several measures of the performance of a filter [Haykin]. These include rate of convergence, robustness, computational requirements, numerical properties, and how closely the filter reproduces the ideal Wiener filter for cases where the noise and signal statistics are well characterized. Another way to characterize performance is to use the filter to process the real data stream to feed a decision algorithm that must extract information from the data. This provides a measure of how much information the filter is pulling from the data.

The recursive exponential filter was used to filter raw aerosol concentration measurements for input into a non-ambient aerosol detection algorithm. The data was collected outdoors at Dugway Proving Ground in Utah during September 1997, and again during September, 1998. The data consists of particle concentrations in each of six particle size ranges (1 to 2, 2-4, 4-6, 6-8, 8-10, and 10+  $\mu\text{m}$ ). This data was collected in 9 second samples, taken every 10 seconds. There were 12 of these particle counters, distributed over a square mile of desert. Data

was collected during normal conditions, in which either the ambient background aerosols, or the aerosols generated by normal local activity (vehicular) were observed. In addition, data was collected after upwind release of non-local, non-ambient aerosol. In total there were approximately 2000 total sensor-hours of background aerosol data collection, and 71 releases of non-ambient aerosol.

A detection algorithm received the data as input and classified the situation as ambient or non-ambient. This algorithm extracts the information from the data stream of whether or not there is non-ambient aerosol present. The algorithm uses filtered data streams, and looks for changes in the total concentration and in the particle size distribution function. The algorithm is adaptive, in that it learns the local background particle size distribution during a several hour training period, and then looks for deviations from what belongs in the area.

In this application, concentration fluctuations that occur on timescales of less than a minute or so are likely to be due to background phenomena or observation (shot) noise rather than non-ambient aerosol release. A short time scale filter was thus used to reduce the effects of these fluctuations in the data stream. The time scale of this short filter was taken to be 4 minutes. Concentration fluctuations with very long timescale are also more likely to be due to background phenomena. To remove the effects of these long time scale background phenomena from the data stream, an exponential filter with a one hour time constant was used. Each of the six data channels was filtered to produce two data streams that fed the algorithm: a 4 minute exponential filter and a one hour exponential filter. The contribution to the recent concentration data attributable to the phenomena of interest ( i.e. the artificial release of non-ambient aerosol) is obtained by subtracting the long time scale filtered value from the short time scale filtered value.

In addition to the recursive exponential filter, a Kalman filter and a moving average filter were also implemented for comparison.

## **8. Results**

A comparison test was constructed with the aerosol data. The 15 nights collected in 1997 were divided into two training nights for the adaptive detection/classification algorithm, and the remaining 13 nights were used as test data. Likewise, the 10 nights collected in 1998 were divided into two training nights, and the remaining 8 nights were used as test data. In total, the test contains 53 test releases, separated by a total of approximately 140 hours of background conditions.

When the data stream is filtered with a time-sensitive recursive exponential filter, with a time constant of four minutes, the classification algorithm successfully identifies 39 of the 53 releases, and never gives a false alarm during non-release conditions.

A Kalman filter was also evaluated against the same test data set, feeding the same classification algorithm. The Kalman parameters were selected so that  $R = 552 Q$ . The Kalman gain and the Kalman variance update equations are combined into a single equation that only requires the ratio of  $R/Q$ , and does not depend on their absolute magnitudes. This leads to a steady-state Kalman gain of  $K=0.0416$ , which is the same gain attained by the time-sensitive recursive exponential filter after a long uninterrupted stream of valid data. With this filter,

the classifier successfully identifies 37 of the 53 releases, but on one occasion falsely classifies a background aerosol event as a non-background event.

A third filter was also evaluated on the same data set: an eight minute moving average. To treat cases of data drop-out, a timestamp was attached to each data record, and the moving average included only those data with a timestamp within eight minutes of the current time. Two validity measures were applied: statistical significance was imposed by requiring each filtered channel to have at least 0.1 particles per liter of air, and no more than 10% of the previous eight minutes of data could be missing. When these conditions are not met, the filtered value is marked invalid, and not used by the classifier. When the detection/classification algorithm was fed with this filtered data, the resulting performance was successful detection of 35 of the 53 releases, with no background aerosol events falsely classified as non-background events. Moving average filters with several other durations were also evaluated: eight minutes was optimal.

## **9. Conclusion**

The time-sensitive recursive exponential filter formulation is simpler to implement and faster to evaluate than a non-recursive filter such as a moving average. It is more robust to data drop-out than a moving average, because it gives age-appropriate weight to missing data in estimating the validity of the filtered result. The exponential filter was found to enable superior performance in an adaptive detection/classification algorithm relative to the moving average filter, boosting performance from 35 to 39 release detections.

The time-sensitive recursive exponential filter was also found to perform somewhat better than a basic Kalman filter having no explicit provision for treatment of data drop-out. The time-sensitive recursive exponential filter can be viewed as a Kalman filter with a particular formulation for the growth of uncertainty with time due to the process noise. This formulation is appropriate for processes wherein the measurement noise (including a high-frequency process noise component), and the low-frequency part of the process noise are uncharacterized and non-stationary. This filter was found to be effective for a detection and classification application with natural and artificial aerosol concentration and particle size profiles.

## **10. References**

- Kalman, R.E. "A new approach to linear filtering and prediction problems," 1960. Trans. AMSA Journal of Basic Engineering, vol. 82, pp 35-45.
- Kalman, R.E. and Bucy, R.S., 1961. "New results in linear filtering and prediction theory," Trans. AMSA Journal of Basic Engineering, vol. 82, pp 95-108.
- Haykin, S., 1986. Adaptive filter theory, Prentice-Hall Information and System Science Series.
- Press, W.H., et. al., 1992. Numerical Recipes in C, Cambridge University Press.